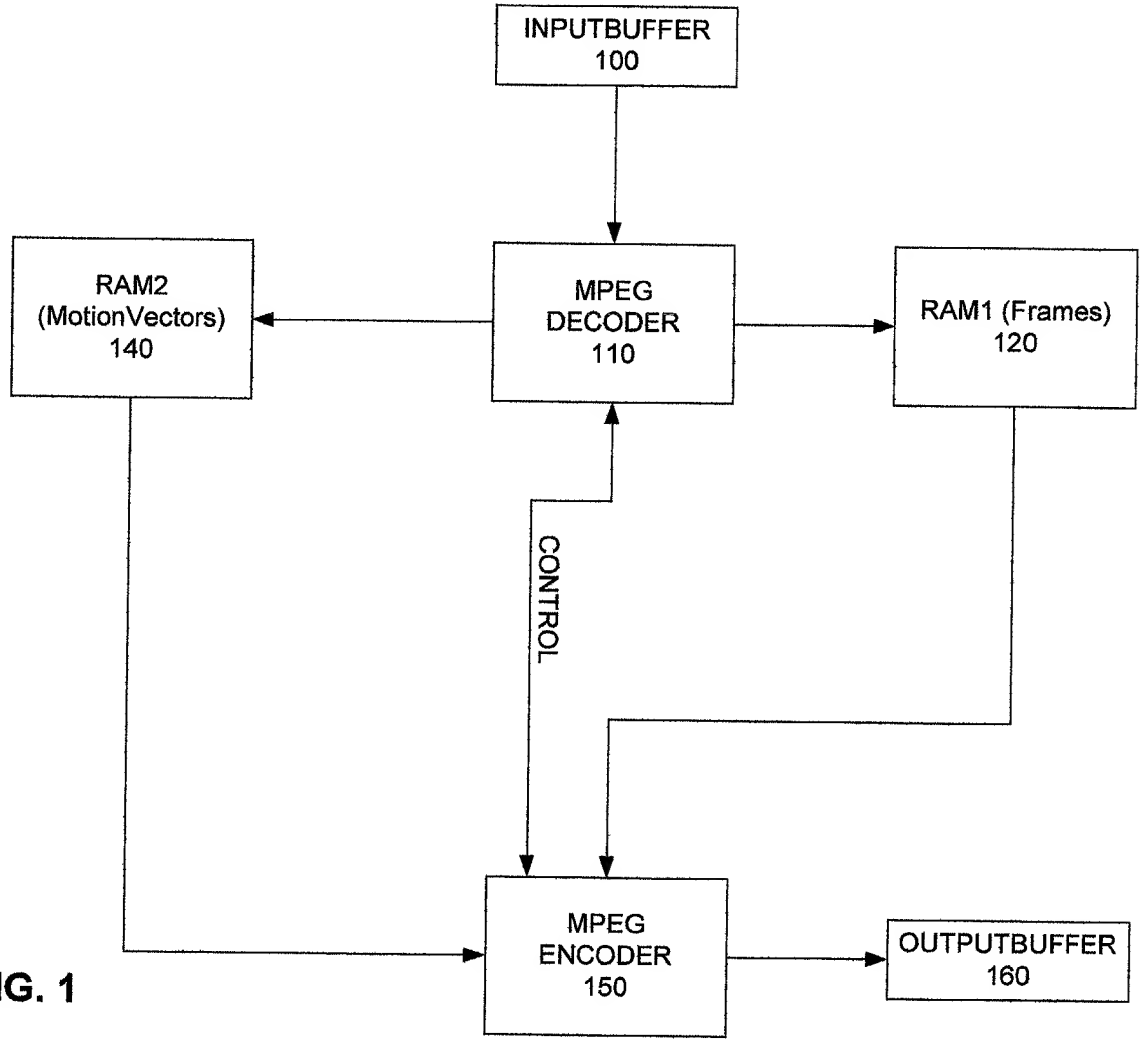
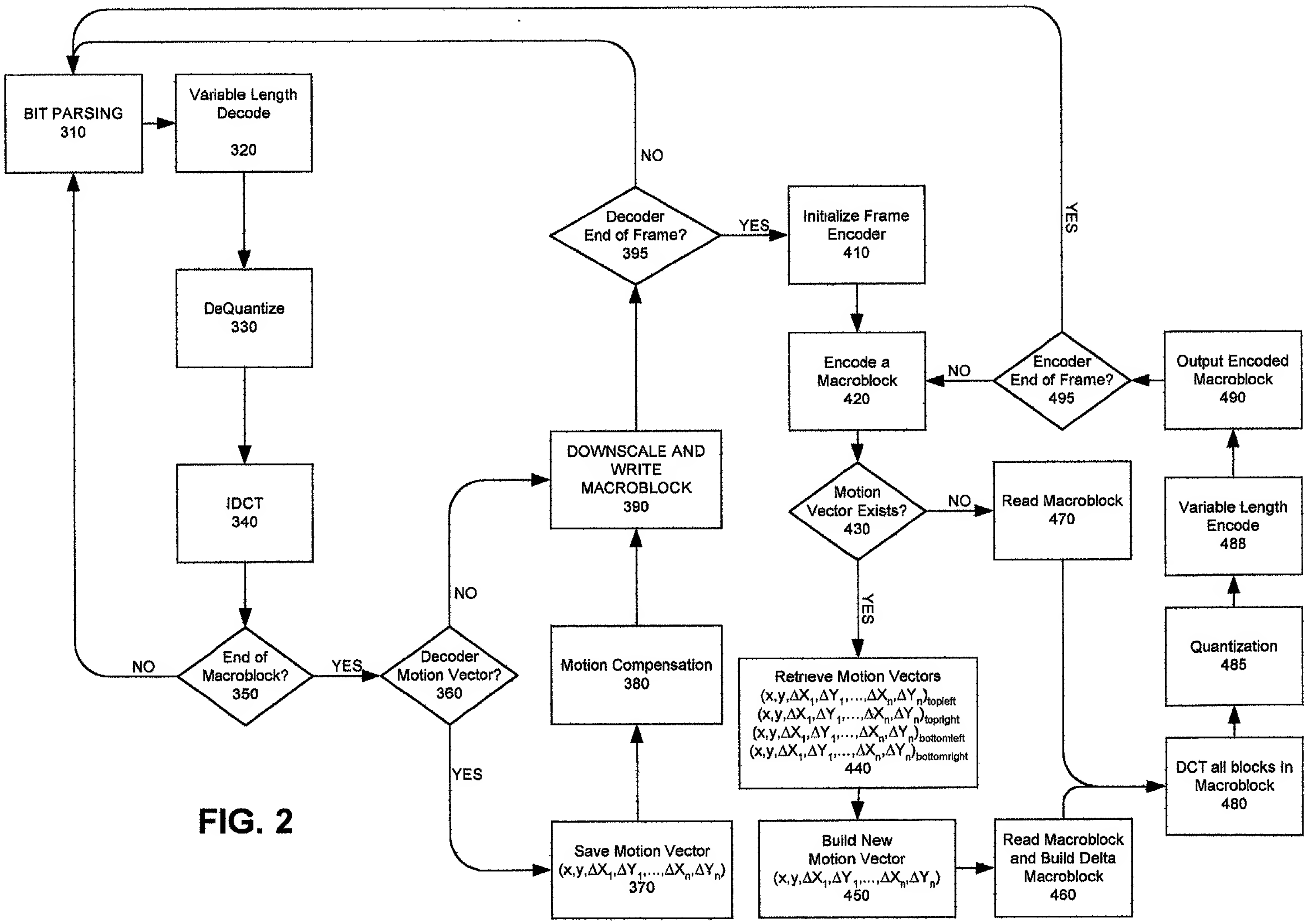


FIG. 1



BIT PARSING 310
Variable Length Decode 320
DeQuantize 330
IDCT 340
End of Macroblock? 350
NO
YES
Decoder Motion Vector? 360
NO
YES
Save Motion Vector (x,y,ΔX1,ΔY1,...,ΔXn,ΔYn) 370
Motion Compensation 380
DOWNSCALE AND WRITE MACROBLOCK 390
Decoder End of Frame? 395
NO
YES
Initialize Frame Encoder 410
Encode a Macroblock 420
Motion Vector Exists? 430
NO
YES
Retrieve Motion Vectors
(x,y,ΔX1,ΔY1,...,ΔXn,ΔYn)topleft
(x,y,ΔX1,ΔY1,...,ΔXn,ΔYn)topright
(x,y,ΔX1,ΔY1,...,ΔXn,ΔYn)bottomleft
(x,y,ΔX1,ΔY1,...,ΔXn,ΔYn)bottomright
440
Build New Motion Vector (x,y,ΔX1,ΔY1,...,ΔXn,ΔYn) 450
Read Macroblock and Build Delta Macroblock 460
Read Macroblock 470
DCT all blocks in Macroblock 480
Quantization 485
Variable Length Encode 488
Output Encoded Macroblock 490
Encoder End of Frame? 495
NO
YES

FIG. 2

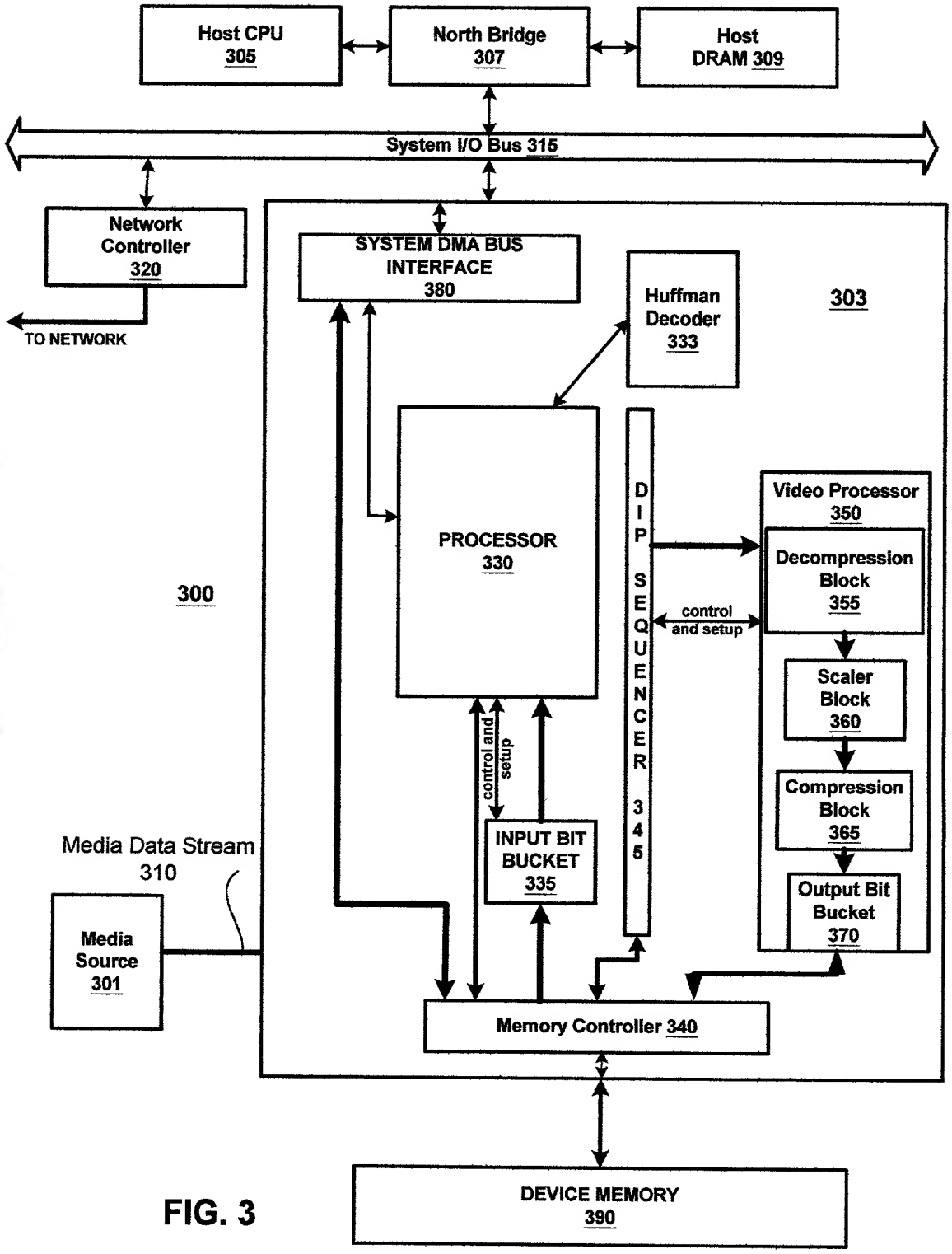

Host CPU
305
North Bridge
307
Host
DRAM 309
System I/O Bus 315
Network
Controller
320
TO NETWORK
SYSTEM DMA BUS
INTERFACE
380
Huffman
Decoder
333
303
PROCESSOR
330
DIP SEQUENCER 345
control
and setup
Video Processor
350
Decompression
Block
355
Scaler
Block
360
Compression
Block
365
Output Bit
Bucket
370
300
control and
setup
INPUT BIT
BUCKET
335
Media Data Stream
310
Media
Source
301
Memory Controller 340
DEVICE MEMORY
390

FIG. 3

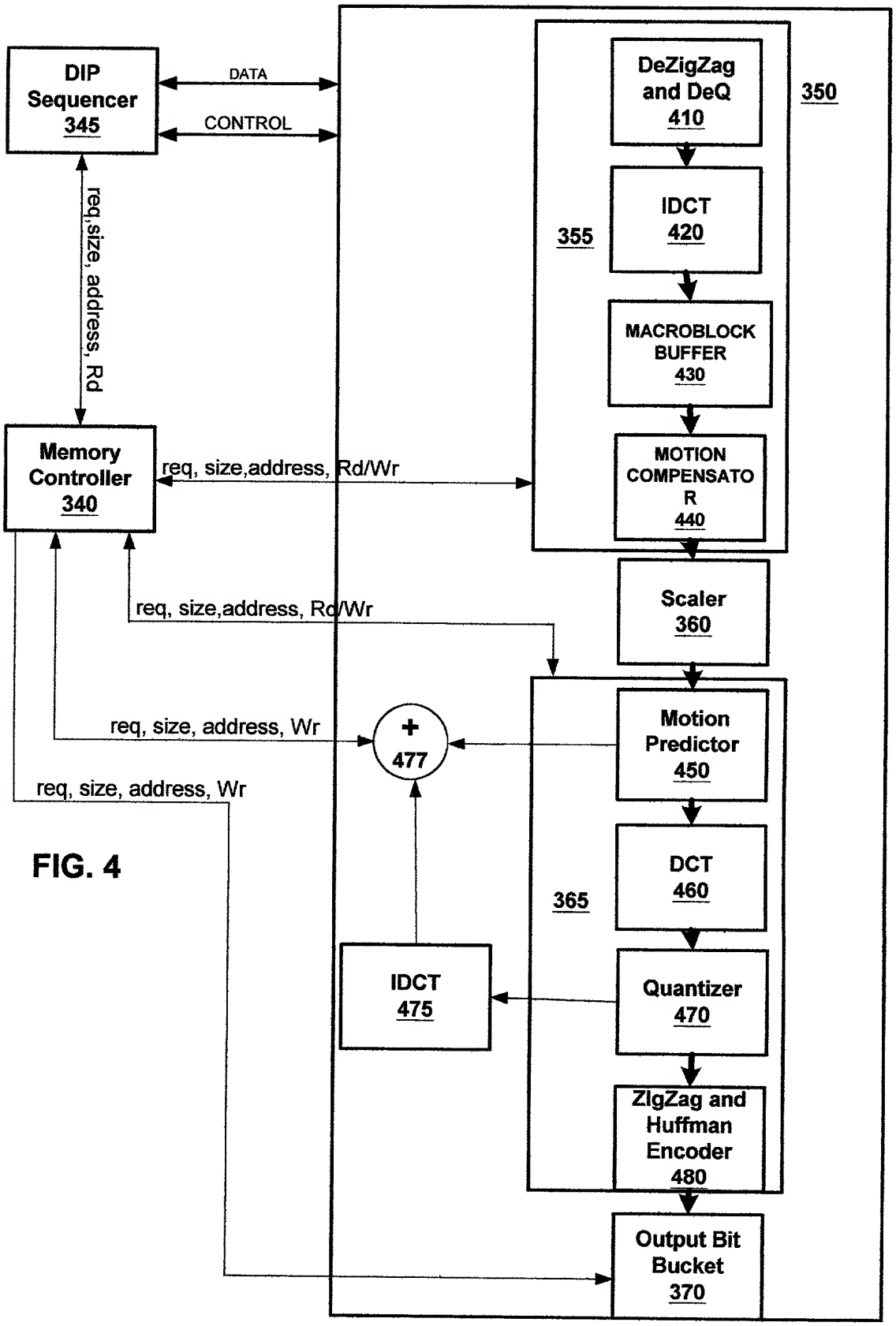

DIP Sequencer 345
DATA
CONTROL
req,size, address, Rd
Memory Controller 340
req, size,address, Rd/Wr
req, size,address, Rd/Wr
req, size, address, Wr
req, size, address, Wr
+ 477
IDCT 475
350
DeZigZag and DeQ 410
355
IDCT 420
MACROBLOCK BUFFER 430
MOTION COMPENSATOR 440
Scaler 360
Motion Predictor 450
365
DCT 460
Quantizer 470
ZIgZag and Huffman Encoder 480
Output Bit Bucket 370

FIG. 4

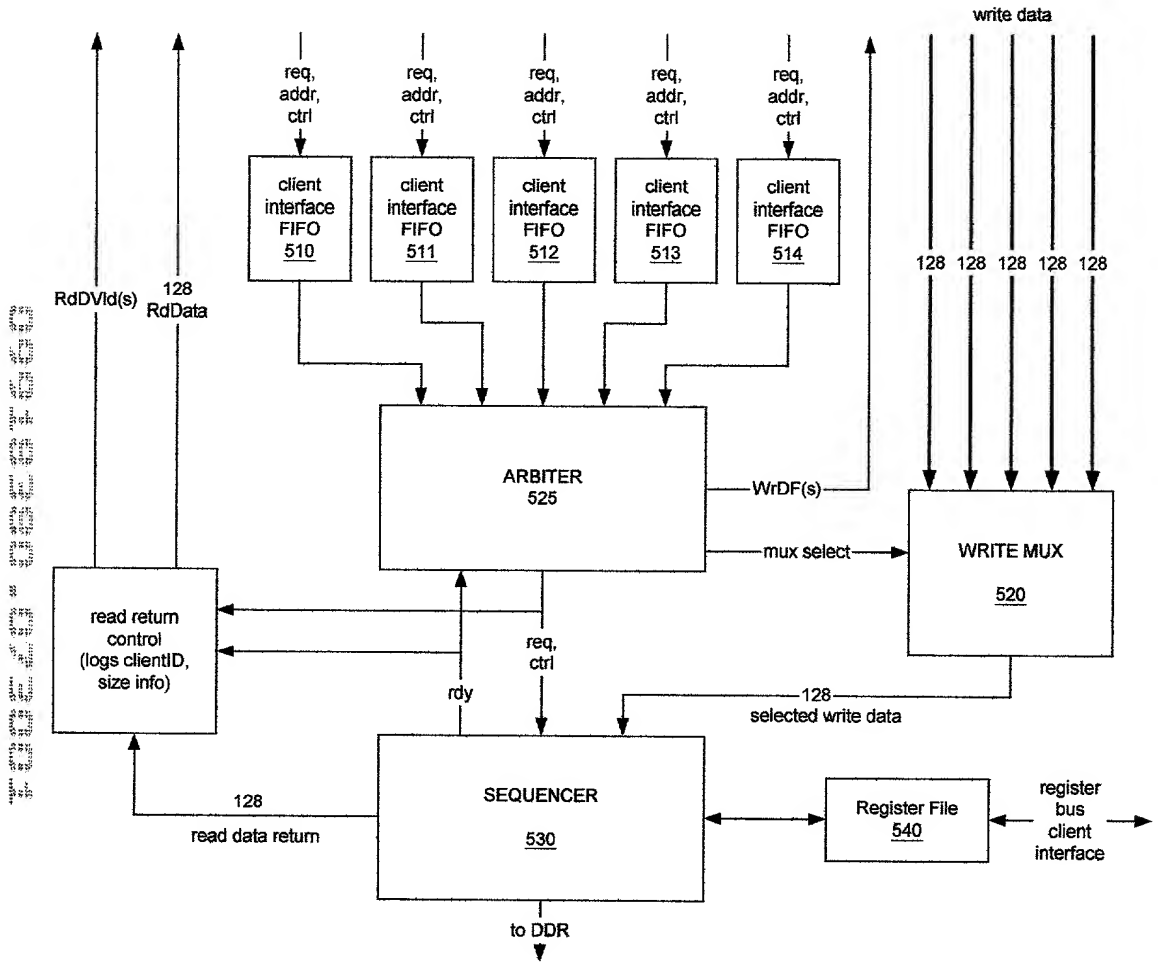

write data
req, addr, ctrl
client interface FIFO 510
client interface FIFO 511
client interface FIFO 512
client interface FIFO 513
client interface FIFO 514
128 128 128 128 128
RdDVld(s)
128 RdData
ARBITER 525
WrDF(s)
mux select
WRITE MUX 520
read return control (logs clientID, size info)
rdy
req, ctrl
128 selected write data
SEQUENCER 530
128 read data return
Register File 540
register bus client interface
to DDR

FIG. 5

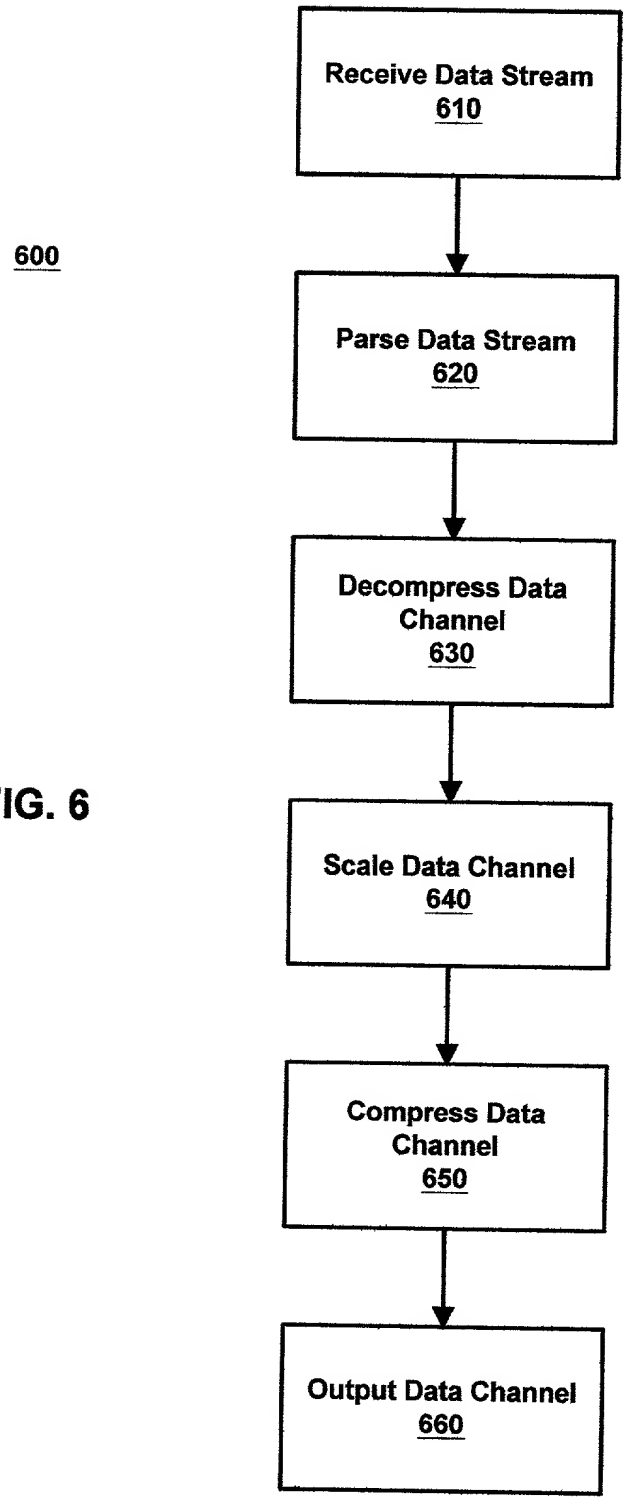

600
Receive Data Stream
610
Parse Data Stream
620
Decompress Data Channel
630
Scale Data Channel
640
Compress Data Channel
650
Output Data Channel
660

FIG. 6

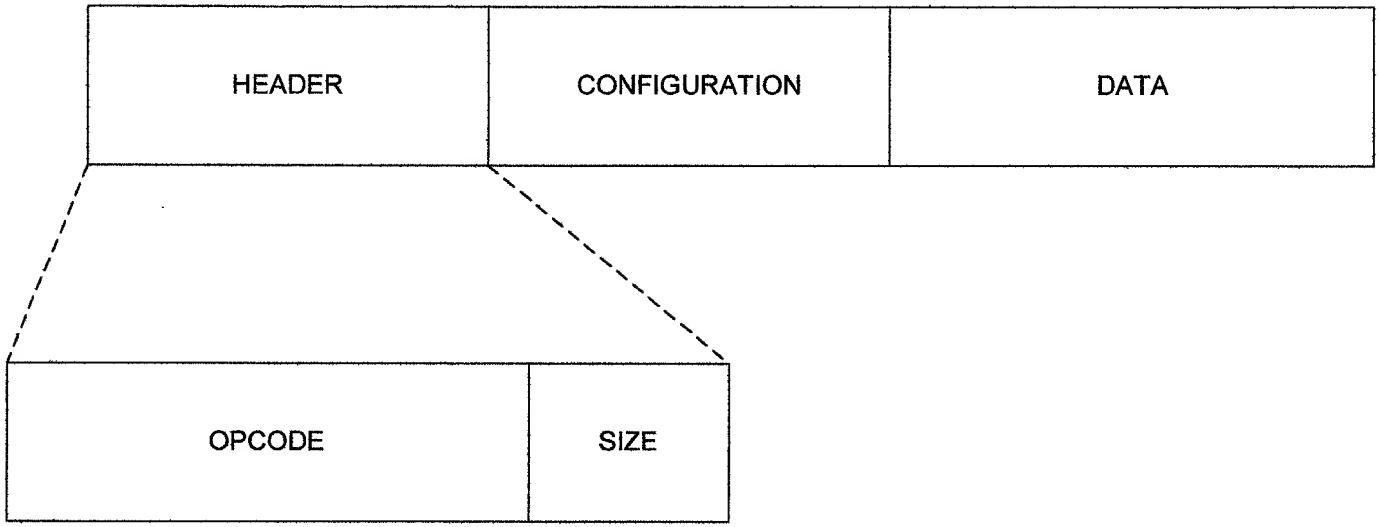


FIG. 7

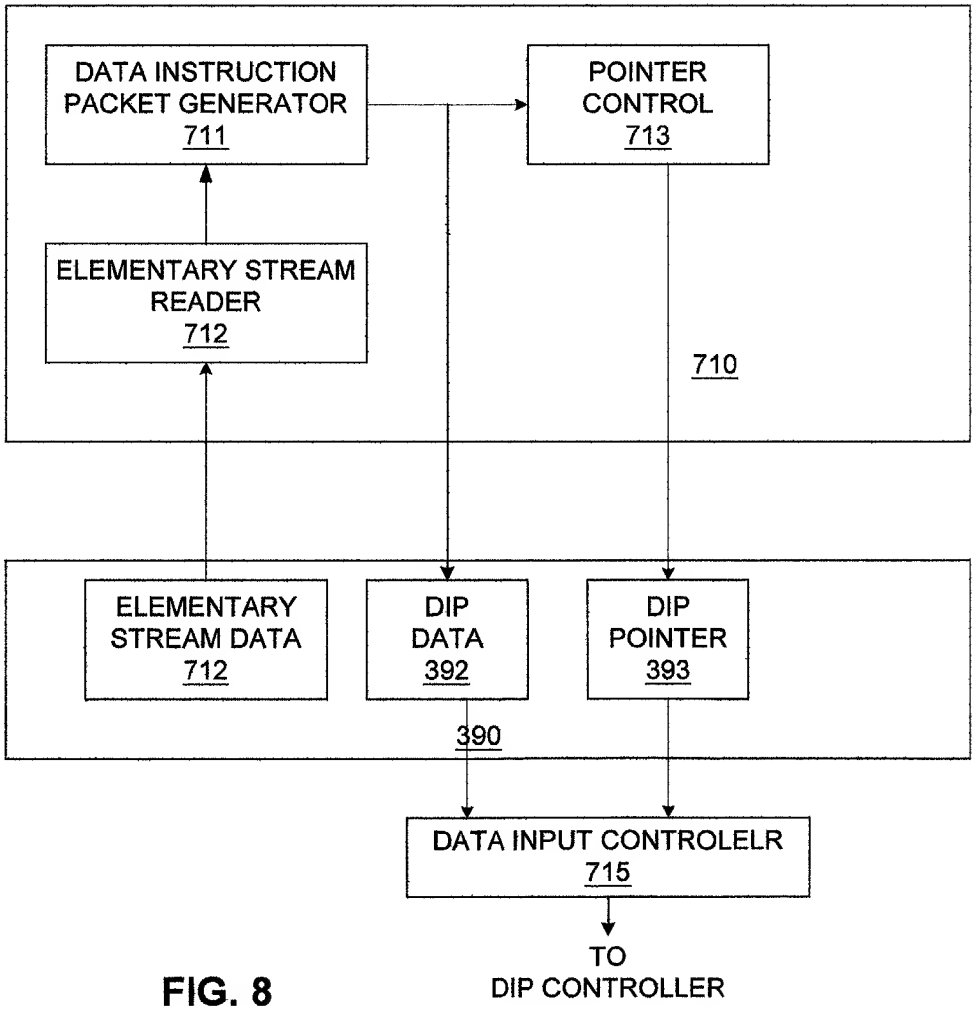


FIG. 8

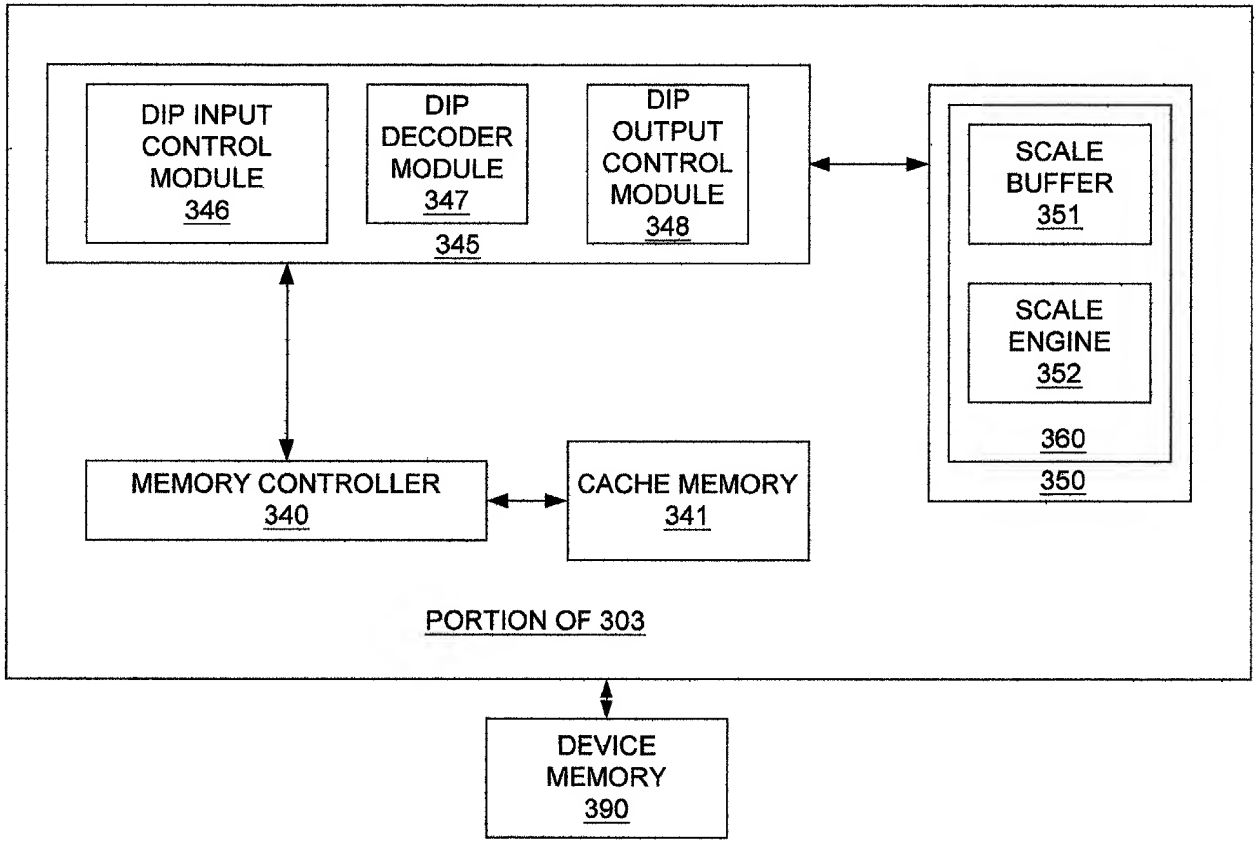


FIG. 9

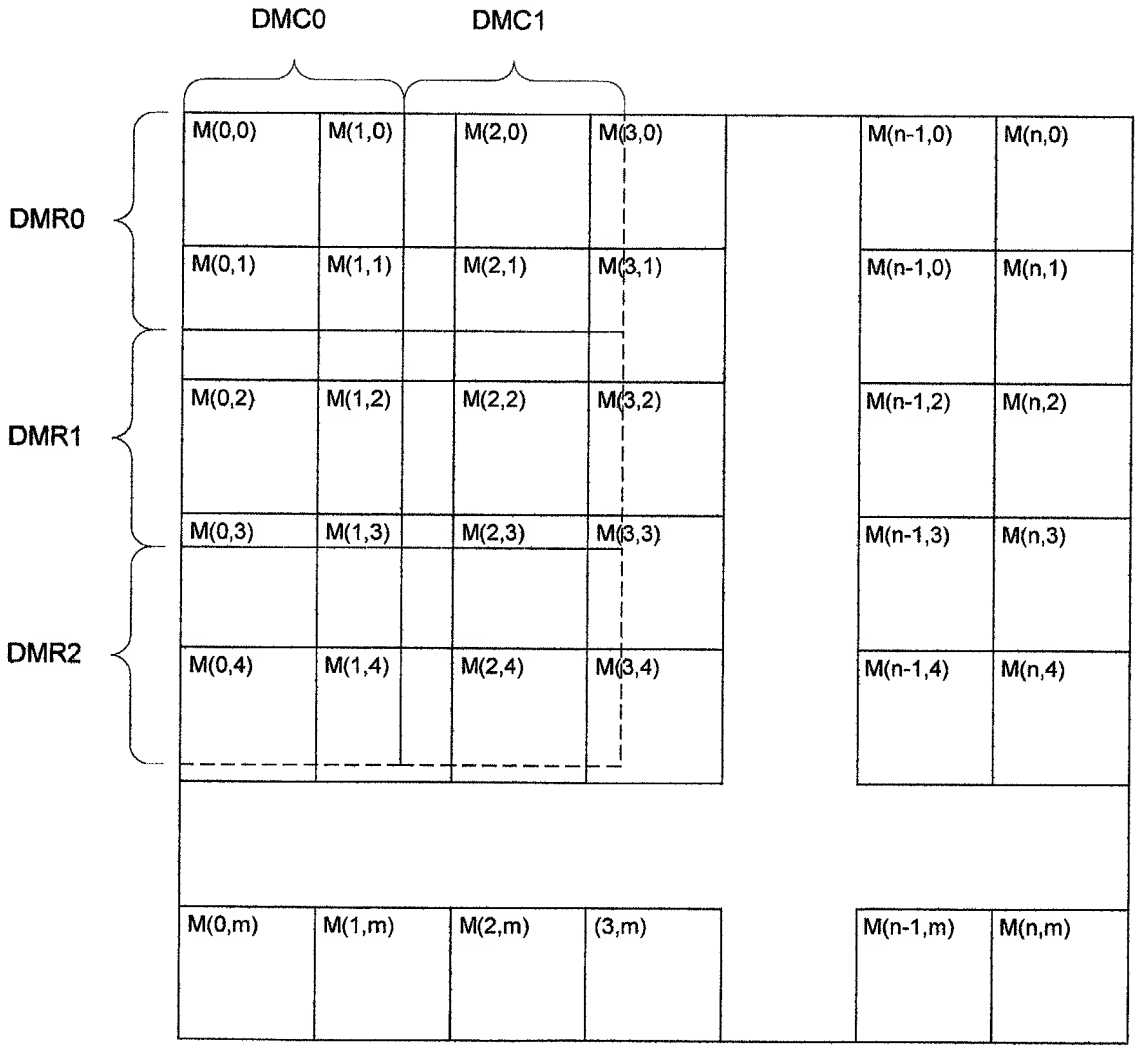


FIG. 10

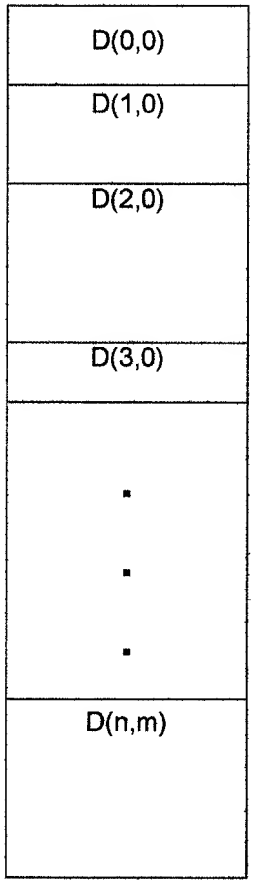


FIG. 11

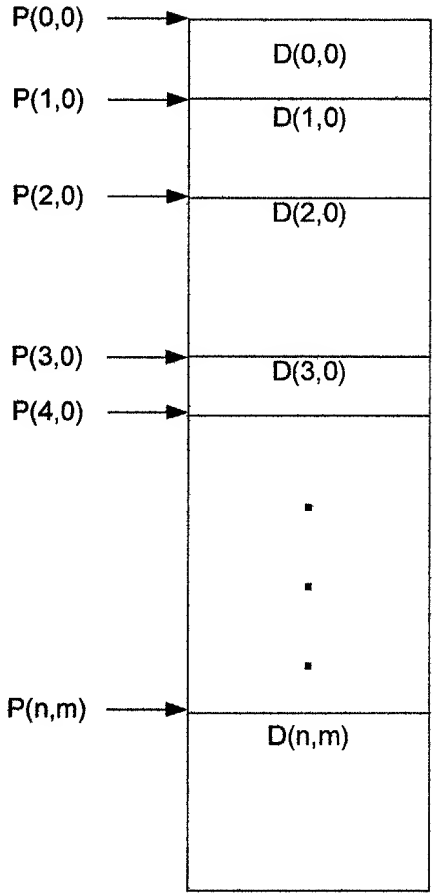


FIG. 13

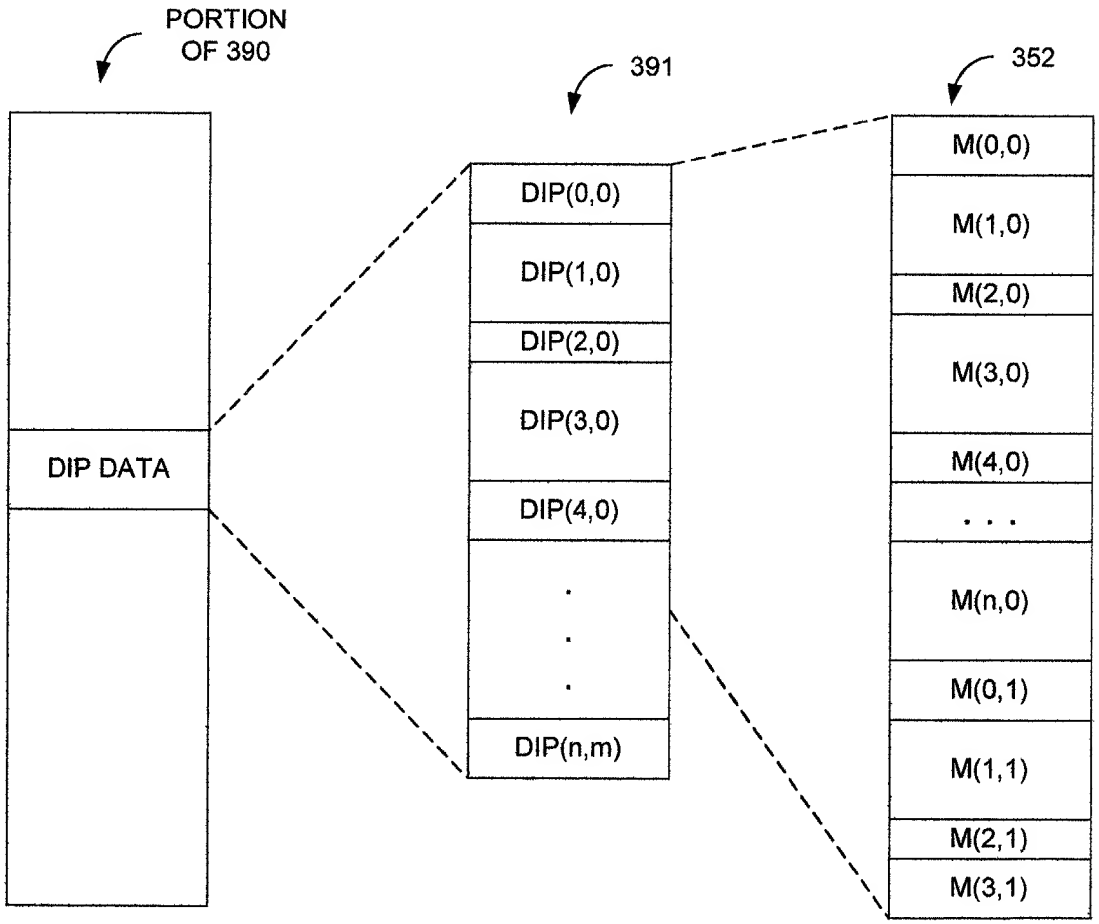

PORTION
OF 390
391
352
DIP DATA
DIP(0,0)
DIP(1,0)
DIP(2,0)
DIP(3,0)
DIP(4,0)
DIP(n,m)
M(0,0)
M(1,0)
M(2,0)
M(3,0)
M(4,0)
M(n,0)
M(0,1)
M(1,1)
M(2,1)
M(3,1)

FIG. 12

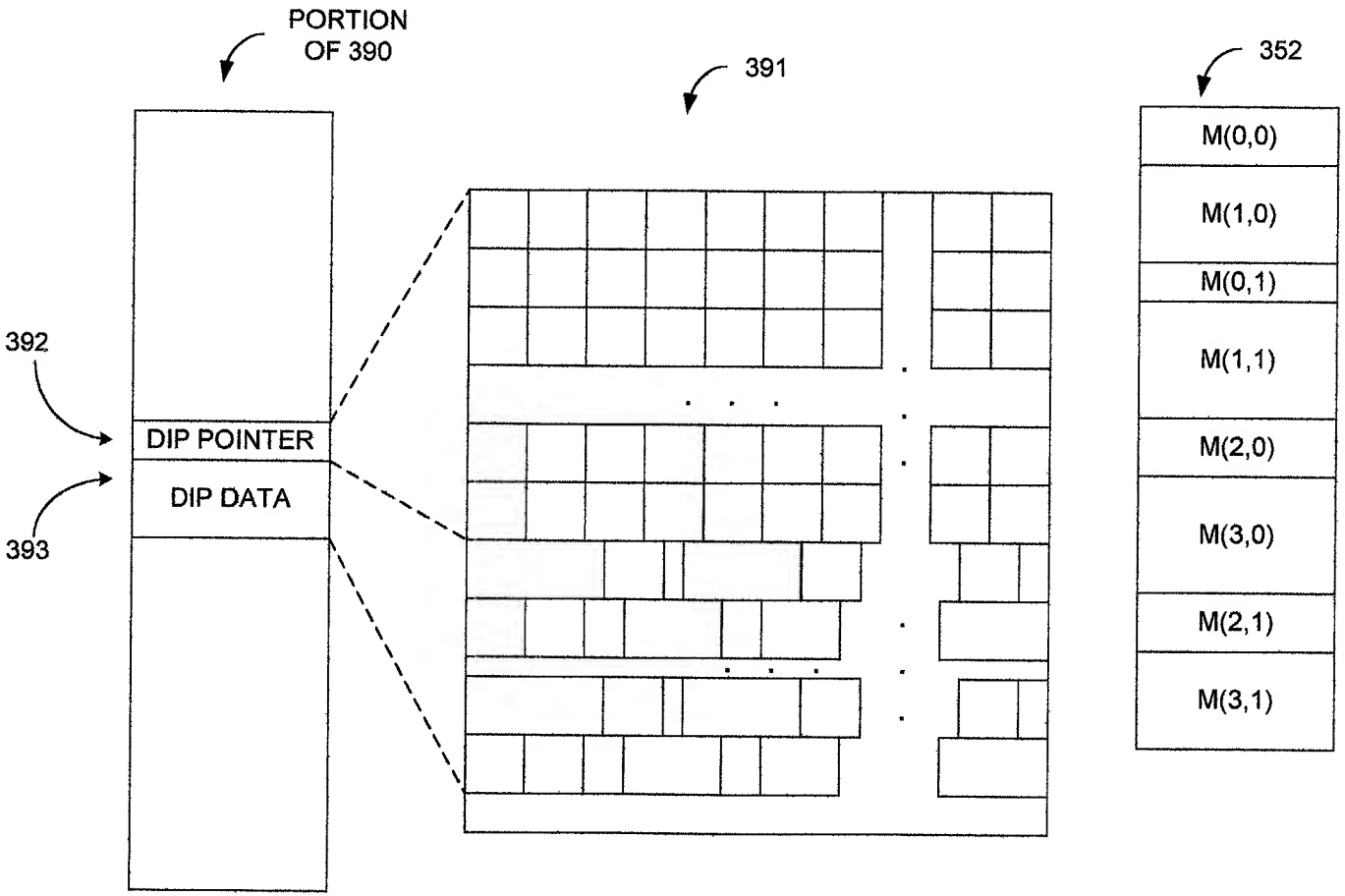

PORTION
OF 390
391
352
392
393
DIP POINTER
DIP DATA
M(0,0)
M(1,0)
M(0,1)
M(1,1)
M(2,0)
M(3,0)
M(2,1)
M(3,1)

FIG. 14

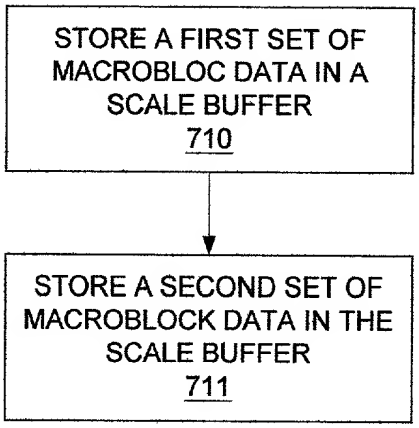

STORE A FIRST SET OF MACROBLOC DATA IN A SCALE BUFFER
710
STORE A SECOND SET OF MACROBLOCK DATA IN THE SCALE BUFFER
711

FIG. 15

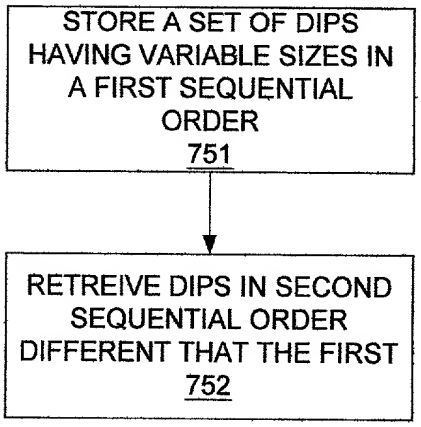

STORE A SET OF DIPS HAVING VARIABLE SIZES IN A FIRST SEQUENTIAL ORDER
751
RETREIVE DIPS IN SECOND SEQUENTIAL ORDER DIFFERENT THAT THE FIRST
752

FIG. 19

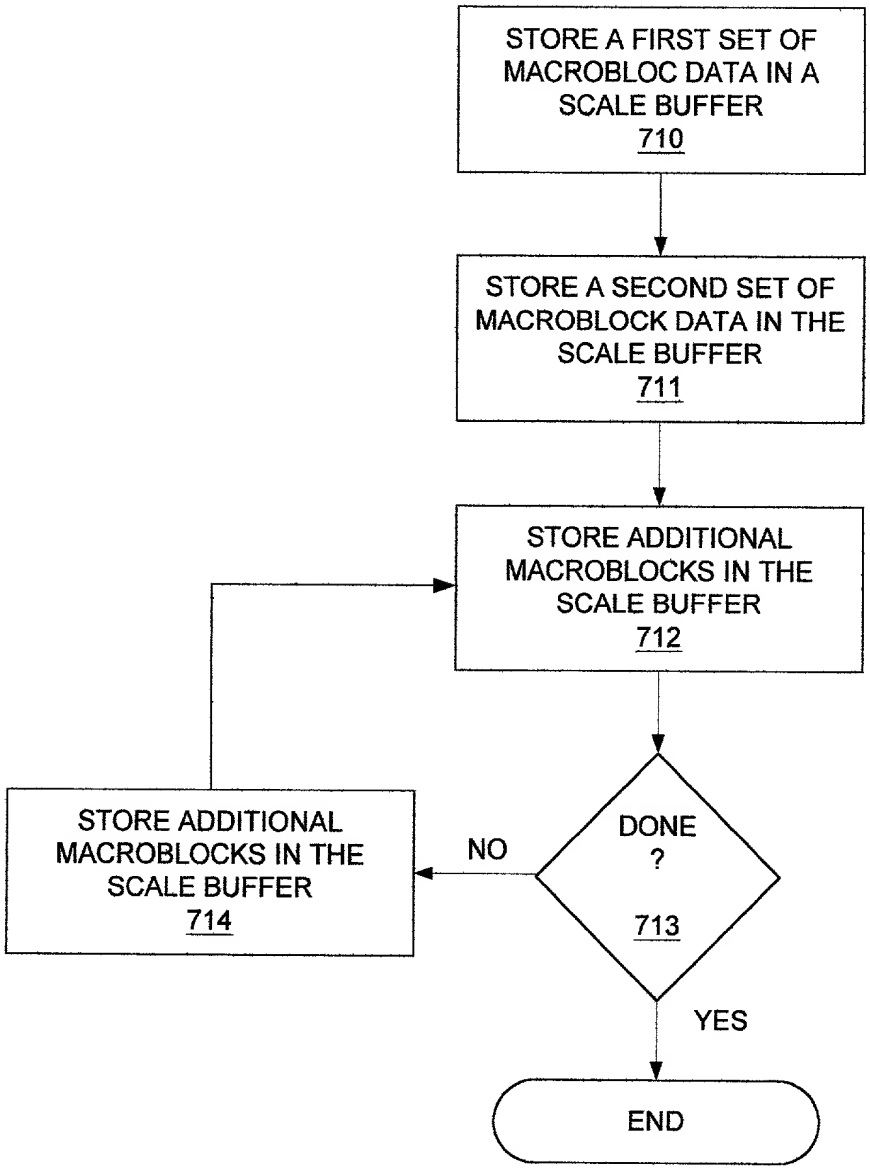

STORE A FIRST SET OF MACROBLOC DATA IN A SCALE BUFFER
710
STORE A SECOND SET OF MACROBLOCK DATA IN THE SCALE BUFFER
711
STORE ADDITIONAL MACROBLOCKS IN THE SCALE BUFFER
712
DONE ?
713
NO
STORE ADDITIONAL MACROBLOCKS IN THE SCALE BUFFER
714
YES
END

FIG. 16

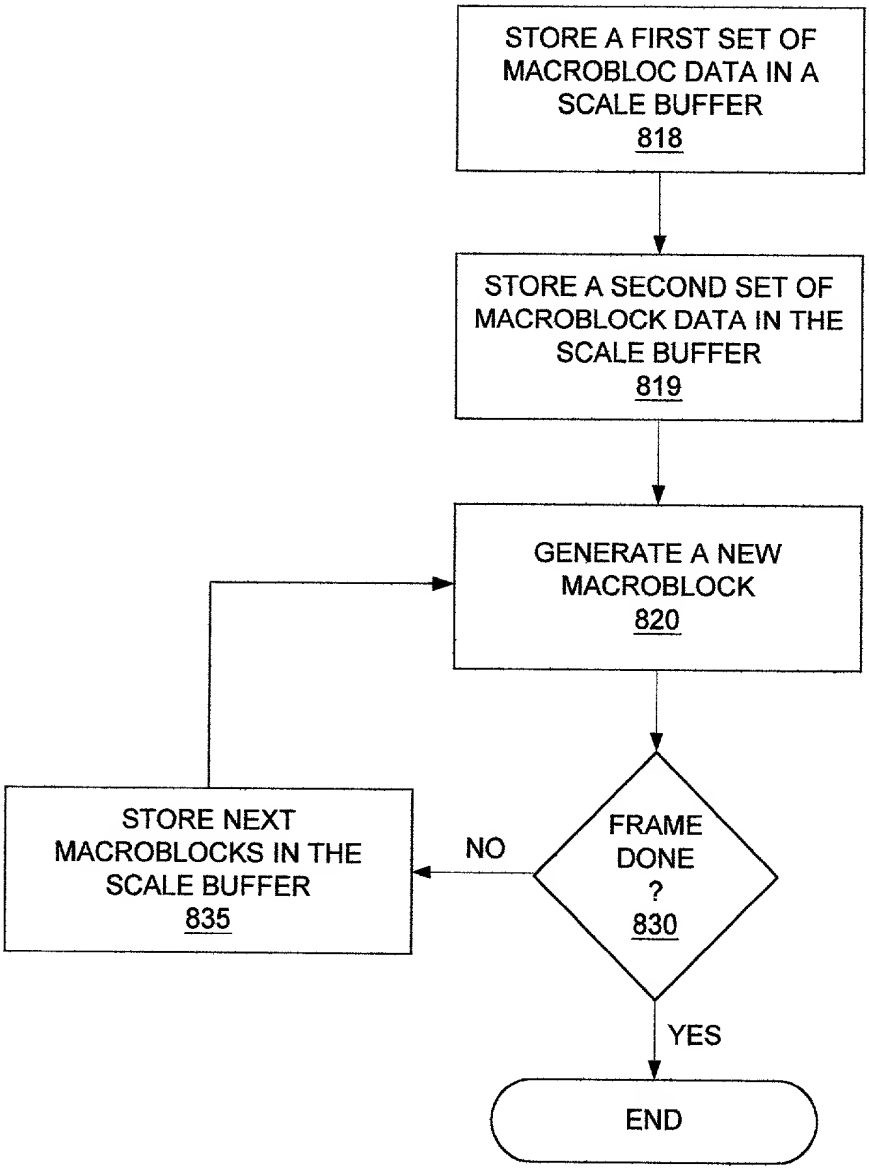

STORE A FIRST SET OF MACROBLOC DATA IN A SCALE BUFFER
818
STORE A SECOND SET OF MACROBLOCK DATA IN THE SCALE BUFFER
819
GENERATE A NEW MACROBLOCK
820
FRAME DONE ?
830
NO
STORE NEXT MACROBLOCKS IN THE SCALE BUFFER
835
YES
END

FIG. 17

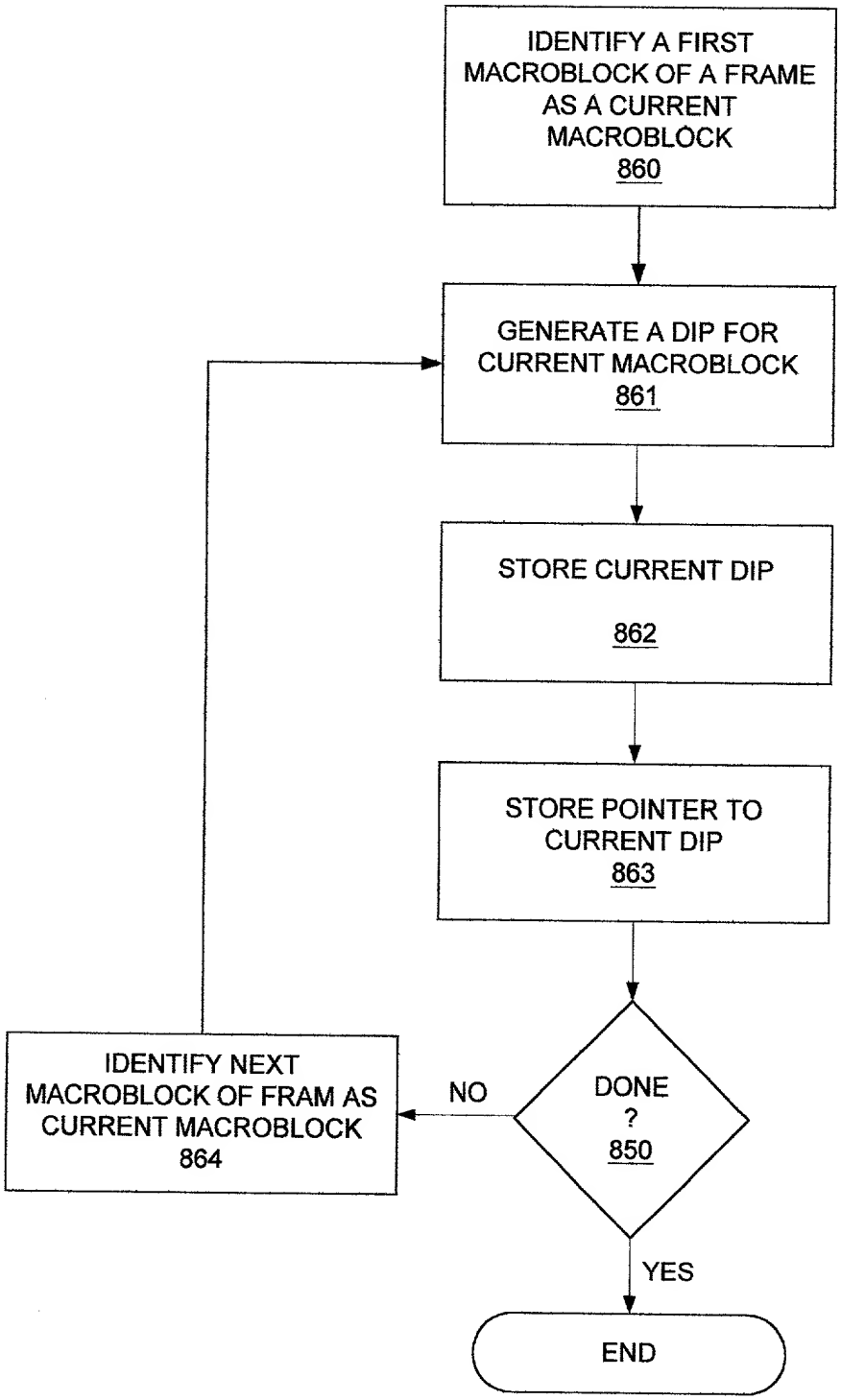

IDENTIFY A FIRST MACROBLOCK OF A FRAME AS A CURRENT MACROBLOCK
860
GENERATE A DIP FOR CURRENT MACROBLOCK
861
STORE CURRENT DIP
862
STORE POINTER TO CURRENT DIP
863
DONE ?
850
NO
IDENTIFY NEXT MACROBLOCK OF FRAM AS CURRENT MACROBLOCK
864
YES
END

FIG. 18

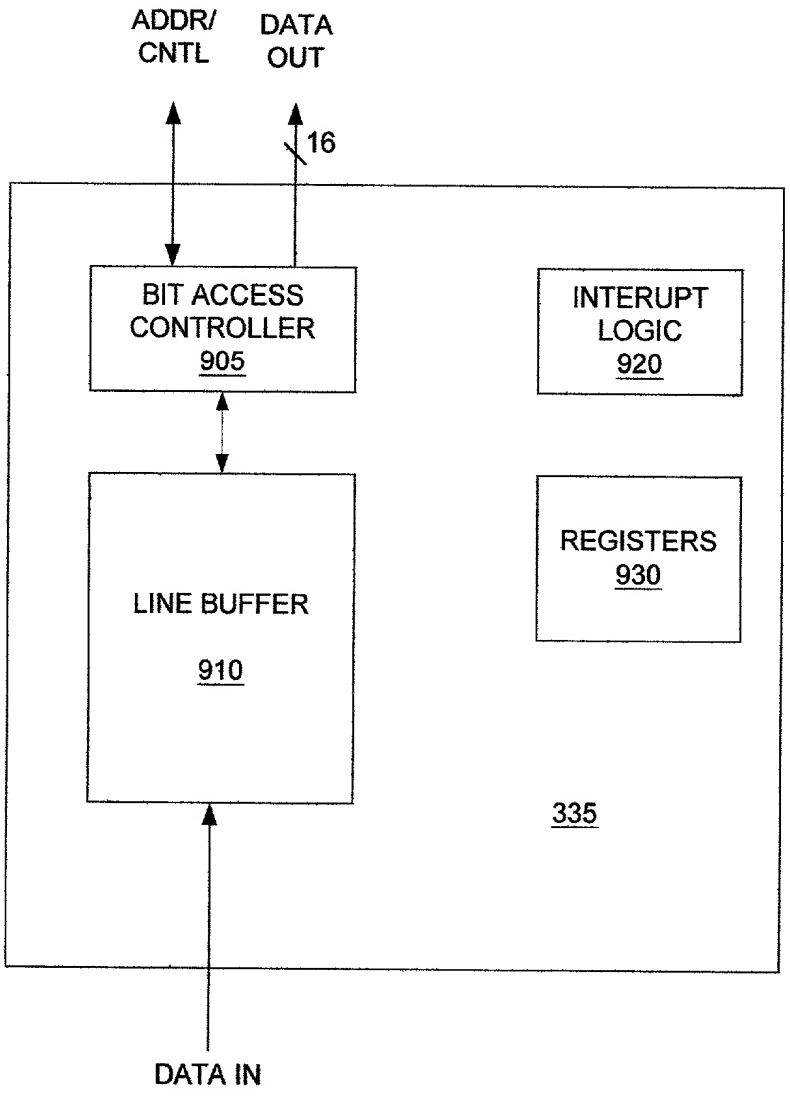

ADDR/
CNTL
DATA
OUT
16
BIT ACCESS
CONTROLLER
905
INTERUPT
LOGIC
920
LINE BUFFER
910
REGISTERS
930
335
DATA IN

FIG. 20

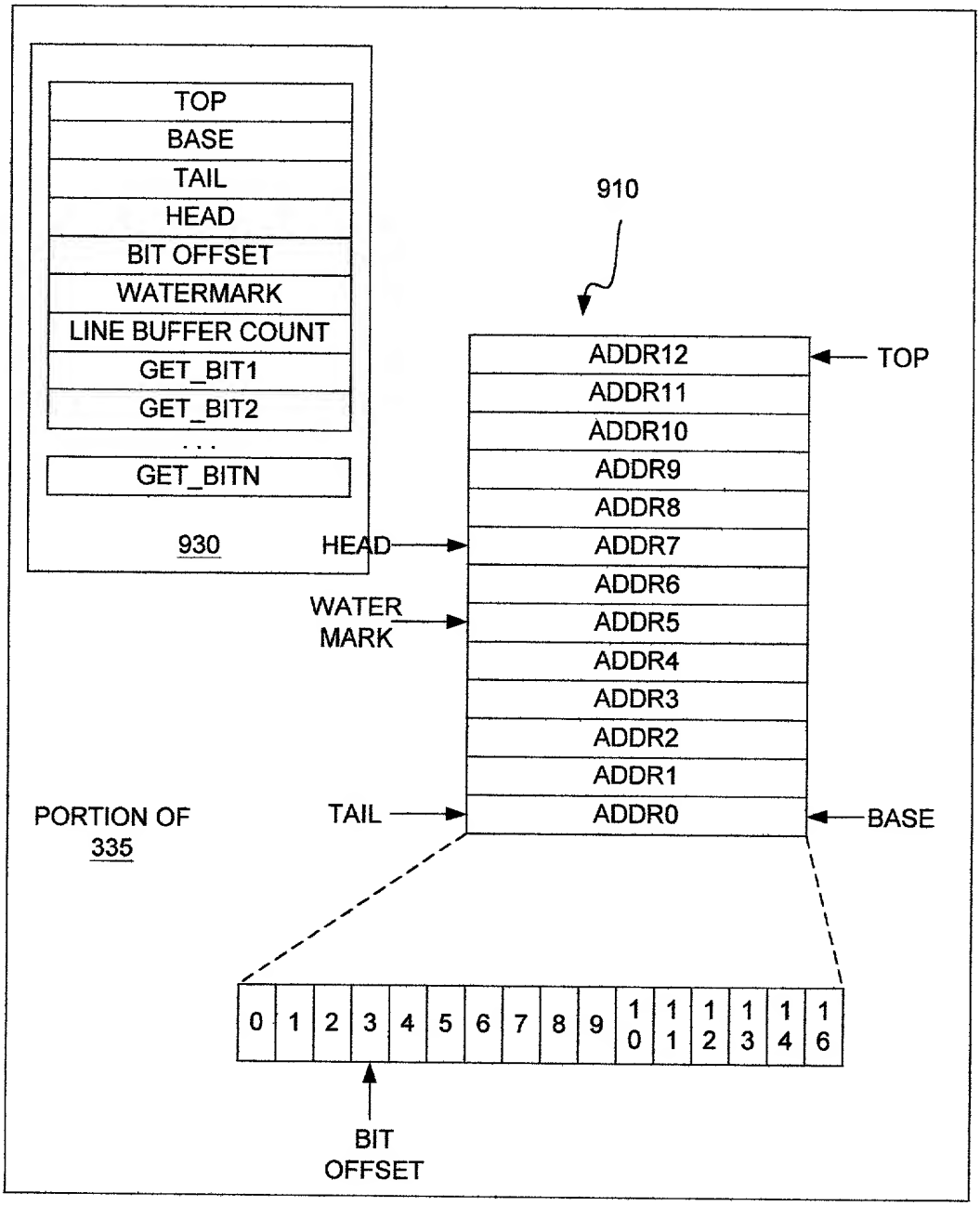

TOP
BASE
TAIL
HEAD
BIT OFFSET
WATERMARK
LINE BUFFER COUNT
GET_BIT1
GET_BIT2
. . .
GET_BITN
930
910
ADDR12
ADDR11
ADDR10
ADDR9
ADDR8
ADDR7
ADDR6
ADDR5
ADDR4
ADDR3
ADDR2
ADDR1
ADDR0
TOP
BASE
HEAD
WATER MARK
TAIL
PORTION OF 335
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 16
BIT OFFSET

FIG. 21

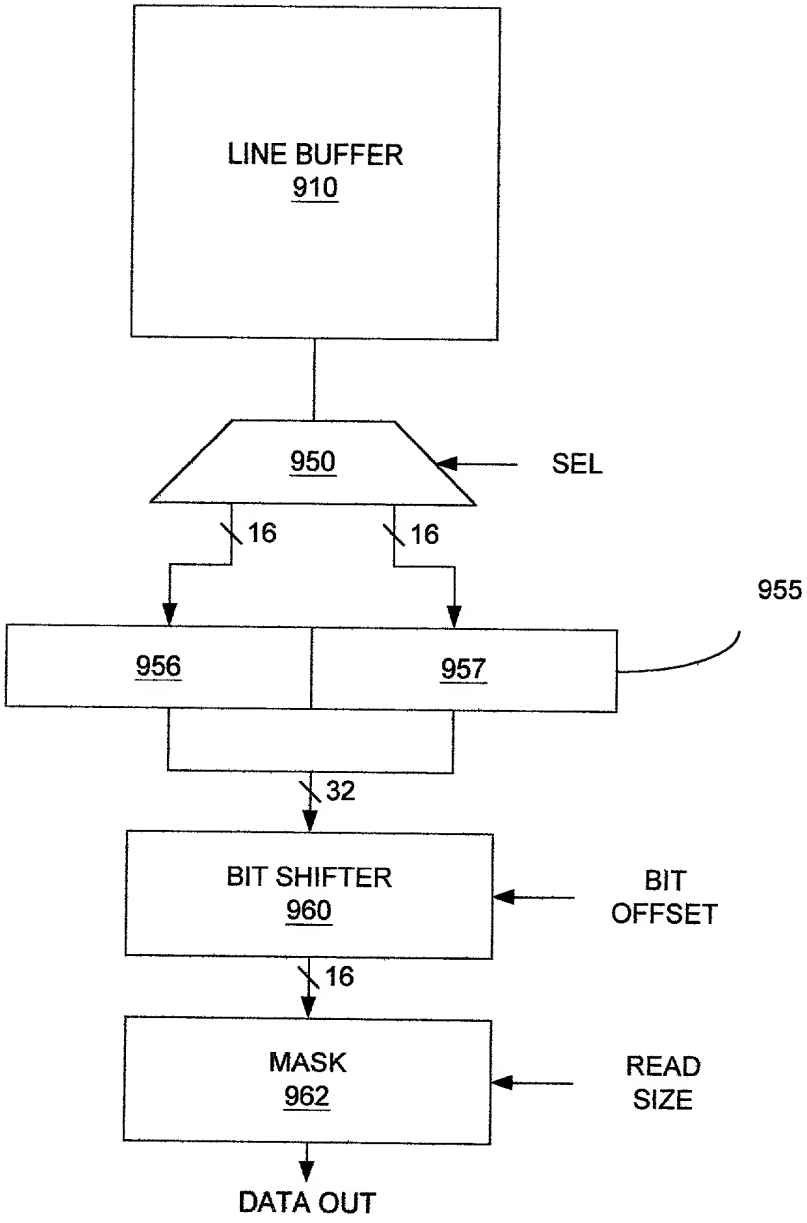

LINE BUFFER
910
950
SEL
16
16
955
956
957
32
BIT SHIFTER
960
BIT
OFFSET
16
MASK
962
READ
SIZE
DATA OUT

FIG. 22

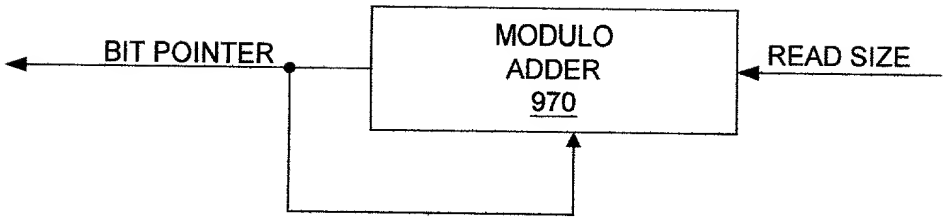


FIG. 23